AD-A225 700

ARI Research Note 90-47

# Theories, Methods, and Tools for the Design of User-Centered Computer Systems

Gerhard Fischer and Walter Kintsch

University of Colorado

for

Contracting Officer's Representative
Michael Drillings

Basic Research
Michael Kaplan, Director

July 1990

United States Army
Research Institute for the Behavioral and Social Sciences

# U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

## A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

| | |
|---|---|
| EDGAR M. JOHNSON<br>Technical Director | JON W. BLADES<br>COL, IN<br>Commanding |

Research accomplished under contract for the Department of the Army

University of Colorado

Technical review by

Michael Drillings

| Accession For | |
|---|---|
| NTIS GRA&I | X |
| DTIC TAB | ☐ |
| Unannounced | ☐ |
| Justification | |
| By | |
| Distribution/ | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |



**NOTICES**

**DISTRIBUTION**: This report has been cleared for release to the Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or the National Technical Information Service (NTIS).

**FINAL DISPOSITION**: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

**NOTE**: The views, opinions, and findings in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other authorized documents.

# THEORIES, METHOD, AND TOOLS FOR THE DESIGN OF USER-CENTERED COMPUTER SYSTEMS

## CONTENTS



LIST OF FIGURES

## CONTENTS (Continued)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>Unclassified | 1b. RESTRICTIVE MARKINGS<br>-- |
| 2a. SECURITY CLASSIFICATION AUTHORITY<br>--<br>2b. DECLASSIFICATION/DOWNGRADING SCHEDULE<br>-- | 3. DISTRIBUTION/AVAILABILITY OF REPORT<br>Approved for public release; distribution is unlimited. |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S)<br>-- | 5. MONITORING ORGANIZATION REPORT NUMBER(S)<br>ARI Research Note 90-47 |
| 6a. NAME OF PERFORMING ORGANIZATION<br>University of Colorado<br>6b. OFFICE SYMBOL (If applicable)<br>-- | 7a. NAME OF MONITORING ORGANIZATION<br>U.S. Army Research Institute<br>Office of Basic Research |
| 6c. ADDRESS (City, State, and ZIP Code)<br>Boulder, CO 80309 | 7b. ADDRESS (City, State, and ZIP Code)<br>5001 Eisenhower Avenue<br>Alexandria, VA 22333-5600 |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Research Institute for the Behavioral and Social Sciences<br>8b. OFFICE SYMBOL (If applicable)<br>PERI-BR | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER<br>MDA903-86-C-0143 |
| 8c. ADDRESS (City, State, and ZIP Code)<br>Office of Basic Research<br>5001 Eisenhower Avenue<br>Alexandria, VA 22333-5600 | 10. SOURCE OF FUNDING NUMBERS<br>PROGRAM ELEMENT NO. 61102B<br>PROJECT NO. 74F<br>TASK NO.<br>WORK UNIT ACCESSION NO. |

11. TITLE (Include Security Classification)
Theories, Methods, and Tools for the Design of User-Centered Computer Systems

12. PERSONAL AUTHOR(S)
Fischer, Gerhard; and Kintsch, Walter

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Year, Month, Day) | 15. PAGE COUNT |
|---|---|---|---|
| Interim | FROM 86/09 TO 89/02 | 1990, July | 41 |

16. SUPPLEMENTARY NOTATION
Contracting Officer's Representative, Michael Drillings

| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB-GROUP | Human-computer interaction, Knowledge representation, Computer programming, Machine learning, |
| | | | |
| | | | |

19. ABSTRACT (Continue on reverse if necessary and identify by block number)


There are many aspects to the term "user-centered." For this research, we have concentrated on the issue of information management. Our theoretical starting point was to identify the distinction between two levels of mental representations that users have of the tasks they want to perform on computer systems. In our proposal to ARI, we planned to distinguish between a situation model and a system model. The situation model is a representation of the task the user wants to perform in terms of the specific task domain. It is subjective and varies somewhat among individuals, but our assumption has been that it is well specified. In order to do anything, however, the user's situation model must be transformed into a system model, which is normative and system specific. This distinction has been the driving idea behind the theorizing and system building in this project. Our question has been, how, for a variety of tasks in which information management plays a central role, this transformation is achieved, and what system support can be provided for it.


| | |
|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS | 21. ABSTRACT SECURITY CLASSIFICATION<br>Unclassified |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL<br>Michael Drillings | 22b. TELEPHONE (Include Area Code)<br>(202) 274-8722<br>22c. OFFICE SYMBOL<br>PERI-BR |

DD Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

# THEORIES, METHODS, AND TOOLS FOR THE DESIGN OF USER-CENTERED COMPUTER SYSTEMS

## 1. Summary

### 1.1 Objectives

The objectives of the project were stated in the original proposal as follows:

> The goal of this research at the general level is to develop theories, methods, and tools for the design of user-centered computer systems, and at the specific level to design, implement, and evaluate a customizable Personalized Intelligent Retrieval System. Our research is based on the basic hypothesis that the following duality exists:
>
> 1. User-centered system design cannot be done and understood without trying to test existing ones, extend existing ones, and design new ones.
> 2. User-centered system design cannot be understood by just doing it; the system building efforts must be based on a deep understanding of the theoretical and methodological issues behind them, derived primarily from cognitive science, and, as far as evaluation is concerned, from human factors / cognitive ergonomics.

Thus, we proposed not just to build systems, but to discover which systems are worth building and on which principled design strategies these systems are based. This goal has remained unchanged. The research strategy which we adopted to achieve these goals was one of convergence: We started with system building on the one side and theoretical development on the other, and worked to bring these two together ever more intimately. By now, this convergence has been largely achieved, and system building, theory development, and experimentation are closely integrated.

### 1.2 Overview of this Report

In Section 2, we will describe the work done in the last 2 1/2 years in this project. Section 3 will describe the plans for the future, which are based on the above objectives, the new insights gained from our work so far, and the identification of a number of important problems in the area of user-centered system design.

In Appendix I, we will summarize some additional information about the project. Appendix II is an assessment of our research efforts with respect to its relevance for ARI by Thomas Mastaglio. Appendix III briefly describes a number of related research activities, which take direct advantage of the results produced by this project in our overall research work at the University of Colorado, Boulder.

## 2. Progress Report

There are many aspects to the term "user-centered." We have concentrated on the issue of information management (see Figure 2-1). Our theoretical starting point was a distinction between two levels of mental representations users have of the tasks they want to perform with the help of computer systems. In our original proposal to ARI, we proposed to distinguish between a situation model on the one hand and a system model on the other. The situation model is a representation of the task the user wants to perform in terms specific to the task domain. It is subjective and varies somewhat among individuals, but our assumption has been that it is well specified (i.e., the users know what they want to do). In order to do anything, however, the user's situation model must be transformed into a system model, which is normative and system specific. This distinction has been the driving idea behind the theorizing and system building in this project. Our question has been, how, for a variety of tasks in which information management plays a central role, this transformation from situation model to system model is achieved, and what system support can be provided for it.

- **Creating Information**
  - with structure
  - without structure
- **Personalization of Information**
- **Retrieval of Existing Information**
- **Comprehension of the Information Found**
- **Updating/Modification/Extending Existing Information**
  - using structural properties
  - exploiting the context of an information seeking process

**Figure 2-1:** Information Management Operations in Complex Information Stores

Complex information stores require a variety of information management operations. Our project began by concentrating on information retrieval aspects, and we are incrementally integrating the other operations in our work.

## 2.1 Guiding Principles

### Situation Model versus System Model

While a review of this issue is not possible within the present report, we do need to point out that this is not an ad hoc distinction, but is based on theoretical developments in other areas of research. The term situation model was coined in [Dijk, Kintsch 83]. Kintsch and Greeno [Kintsch, Greeno 85] in their work on word arithmetic problems introduced the distinction between a person's understanding of the situation described in a word problem in everyday terms, and the mathematization of that situation (there called the arithmetic problem model). The term system model in the present domain corresponds to the problem model in the mathematics word problem domain. A large number of researchers in cognitive science have used the term mental model (both situation and system model are a type of mental model). The Breckenridge Workshop 1988 [Turner 88] was devoted to a systematic exploration of the issues involved here.

Below, two aspects of our project will be described that have to do with the question how situation models are to be transformed into workable system models. Both concern information management. The first is a system that makes the system model transparent - the HELGON system and its descendants (Section 2.2). It involves a large system building effort, experimental evaluations, and theory-based modifications and developments. The second system actually finds the information: It constructs an appropriate system model, given a situational understanding of routine computing tasks (Section 2.3). It has had, so far, primarily an experimental and theoretical focus, but possible future developments for the design of help systems incorporating some of our findings will be discussed.

### The Role of Structure

The role of structure as an important framework for our research work surfaced first during the Breckenridge Workshop 1987 on "Personalized Intelligent Information Systems" [Fischer, Nieper 87].

There was a fair amount of discussion about the question "Is structure desirable: yes or no?" People do not like being forced into generating structures. In early problem solving stages the enforcement of structure may get in the way when people start doing something. Different approaches towards this problem can be taken by various systems: (a) postpone the necessity for creating a structure and require only a minimal amount of structuring; (b) use spatial relationships as the only clue; (c) require extensive structuring mechanisms (e.g., in hypertext systems).

Underlying these choices is an important design tradeoff: We would like to take advantage of many structuring principles to retrieve and use information, but we are much less willing to take the overhead into account which is necessary for generating this structure. We don't want to structure information explicitly, but we do want to retrieve information using structural properties. Additional issues in relationship to structure are: If we have no structure, then there is no need for restructuring. If we impose a structure, which form should it take: hierarchies, inheritance networks, associations, etc.? Is a structure statically given or generated on demand?

## 2.2 Complex Information Stores[1]

Information retrieval systems provide good examples for the problems that arise when a user's general situation model must be expressed in terms of a specific system model. To formulate the right query in a system model can be difficult, because it requires knowledge that people don't always know exists, and are therefore unable to ask for. One can train people to become experts in a particular system, or one can design systems in such a way that they can be used with a minimum of training. What psychological principles might be relevant to guiding such design?

[1] This section is essentially an abbreviated version of [Fischer et al. 89], based on a paper given at the first meeting of the Human-Computer Interaction Consortium at Vail, CO, in January 1989. It will be submitted to a special issue of "Human-Computer Interaction."

In human memory retrieval, two components are distinguished: the construction of a retrieval cue, and the action of that cue on memory. The first is a control process, usually conscious, and frequently requiring a great deal of problem solving. The second is automatic, rapid, and beyond conscious control. Given a particular retrieval cue and a particular memory, predictable outcomes will be observed. Information retrieval systems need to be concerned with both of these components of memory retrieval.

### 2.2.1 HELGON

Most experimental studies of human memory retrieval are concerned with the retrieval component itself and attempt to keep the control processes involved in setting up the retrieval cue as simple as possible. There are, however, some exceptions, e.g., [Williams, Hollan 81] and [Walker, Kintsch 85]. Williams observed that people were using a strategy of retrieval by reformulation (e.g., having retrieved a girl's name when trying to retrieve the names of one's high-school classmates, one might form a new retrieval cue by asking "Now, whom did she date?").

We have built HELGON [Fischer, Nieper-Lemke 89], a prototypical information management system that is based on the retrieval by reformulation paradigm. In HELGON, the original retrieval by reformulation paradigm is extended in two ways: As in RABBIT [Tou et al. 82; Williams 84], users are supported in the incremental construction of a query to the knowledge base, but they have additional possibilities for specifying the query. And HELGON is more than an information *retrieval* system: It allows the addition of new information by using the same framework that is used for the specification of a query.

The first prototype of HELGON has been evaluated and its shortcomings articulated. A video tape was produced. We are currently in the process of constructing the second prototype of this system.

#### Retrieval by Reformulation in HELGON

The user of HELGON constructs a query incrementally. The current query is shown in the **Query** pane, the list of items matching the query in the **Matching Items** pane, and one of the matching items in the **Example of the Matching Items** pane (Figure 2-2). The query consists of categories and attribute restrictions associated with the categories. Categories as well as attribute values can be either "required" or "prohibited." The user does this by selecting them from the screen or a menu of alternative values or by typing them in on the keyboard. When the user makes additions to the query through input on the keyboard, only useful values, that is, values that exist in the knowledge base, are accepted. This prevents the user from imposing a restriction that would by itself lead to no matching items (e.g., because of a typographical error). The system gives help by completing partial input automatically if it is unique or by listing all possibilities that contain the current input as a substring.

Users can create the query top-down by selecting from the category hierarchy display the category that is expected to contain the desired information. But users may not know in what category the information is stored. Therefore, they can also work bottom-up by criticizing the example. A problem of this approach is that, in a large information store, the example given might be too far away from the desired information. Multiple specification techniques, e.g., first narrowing down the information store by selecting categories, then continuing by criticizing examples, are therefore important in dealing with complex information stores.

In summary, HELGON is designed to make retrieval by reformulation simple: It makes it easy for the user

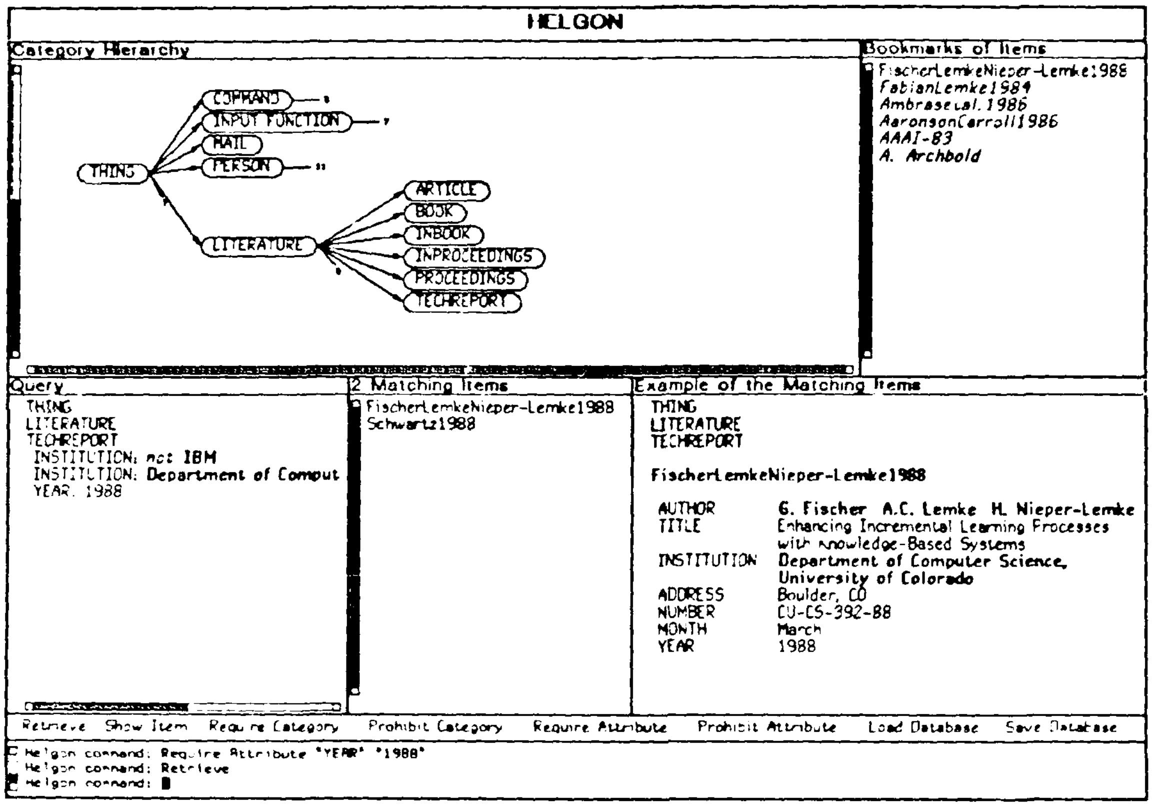


**Figure 2-2:** Literature References in HELGON

to see what the items in the database are like, and to guess how the desired information is to be found by criticizing examples. But it does not force this strategy upon the user: Searching for examples to find the right retrieval cue is a strategy that people use only when more direct retrieval attempts have failed.

## Visualization of the Information Store

It is well-known that users become disoriented in large information stores [Halasz 88]. Therefore, HELGON displays the hierarchical organization of the underlying information store graphically (see the **Category Hierarchy** pane in Figure 2-2). Once users have found a category that seems likely to hold the information they are looking for, they can add it to the query with a mouse click. They can also use the graphical display to edit the hierarchical organization of the information store (e.g., new subcategories can be created).

## Browsing

In addition to assisting the user in defining a query, HELGON supports browsing in the information store. The graphical category hierarchy display can be used to browse categories. Links within the information units can be followed, that is, items that appear as attribute values of other items (displayed in bold face) can be inspected. And items that users looked up previously are added to the **Bookmarks of Items** and allow users to return easily to previous states of their information search.

### Editing by Reformulation

HELGON is not just a tool for *retrieving* information. It allows users to *edit* information and integrates the creation of new knowledge base items with the retrieval by reformulation paradigm. Users first use retrieval by reformulation to find an item that is similar to the new one, and then copy it and use it as a template. In this way, they know which *categories and attributes* are reasonable to use; and because they see examples, they better understand what a category or attribute means. Values can often be reused, or they can be selected from a list of alternative values. They can also be typed in on the keyboard, a feature providing the same support (automatic completion, etc.) as it does in the formulation of a query.

### 2.2.2 The Costs and Benefits of Structure

The database of HELGON is organized hierarchically. This structure significantly reduces the complexity of the information store and makes it easier to retrieve desired items. If we have a well-designed filing system, it is certainly easier to find what we want than if all items are put away randomly. Nevertheless, these benefits of structure are associated with certain costs. There are two such costs: In order to take advantage of the structure of a database, users must learn what that structure is, and in order for a structure to exist in the first place, someone must perform the required encoding.

There are a number of features in HELGON that are designed to help the user find out how the database is structured, and hence, how it can be used. Most importantly, there is the graphical display of the hierarchy that can be expanded and manipulated, and that goes a long way toward teaching the user the nature of the structure.

While these features are very useful, they do impose certain demands on novice users. Indeed, in an empirical evaluation of the system [Foltz, Kintsch 89], novice users eventually solved 92% of the retrieval tasks they were given, but they made a great many errors along the way. On 83% of all tasks at least one error occurred, and a substantial part of these errors were hierarchy errors (30%). Thus, even if users know what the hierarchy is, it is not always easy to work within the constraints it imposes.

There is another type of cost associated with structured databases, and that is the cost of generating that structure in the first place. A system called INFOSCOPE (see Section 3.2.2) explores these costs and attempts to find ways to minimize them. INFOSCOPE is a system which integrates the reading of electronic mail and USENET news into a single information environment including intelligent agents which will help users to impose their own personal structure on the database. Unstructured messages can be structured at the time of reading and storing by agents that have available at least a crude semantics and have observed a particular user's practices and preferences. Thus, at least some the burden would be taken off the user to design and maintain some file or folder structure by having autonomous agents take over structural encoding, at least in part.

The question that remains is what sort of structure is most useful in a database. We know from the psychological literature on learning and memory that structure and organization help, in almost any form, for almost all tasks (e.g., [Kintsch 77]). However, it has to be the right structure for a given task. The more structure the better, if the structure is task appropriate, but if the information is structured the wrong way, performance interference occurs. Complex human performance requires above all flexibility. A personal database must be useful for many different tasks and must be useful over long periods of time.

But the organization appropriate for today is quite different from the one required next year when old projects have terminated and new ones begun. A semantic hierarchy is as hard to reorganize as the proverbial cemetery and/or university. What we need are not semantic hierarchies (as in [Collins, Quillian 69]), or fixed scripts (as in [Schank, Abelson 77]), but memory organization packets of Schank [Schank 82] or associative networks [Kintsch 89].

### 2.2.3 Associative Nets and Associative Retrieval

Several models of human memory exist that do quite a good job at explaining the body of data that has accumulated in psychological laboratories during the last decades. They tell us how human retrieval works. *Given an associative memory and a particular retrieval cue, what will happen?*

Using these models as a basis, it is possible to construct a system that retrieves information from an associative net (corresponding to human long-term memory) via the known processes of human memory retrieval. We chose to model these processes according to the retrieval theory in [Raaijmaker, Shiffrin 81], which needed to be only minimally elaborated to serve for our purposes. Our simulation has these components:

1. an associative database,
2. a control process to set up retrieval cues (retrieval by reformulation is used for this purpose),
3. an automatic retrieval mechanism (after current memory models).

Such a prototype system, dubbed RETRIEVE, has been implemented [Foltz, Kintsch 89].

In an associative net, items enter into many different associative fields, and we don't want to retrieve information from contextually irrelevant fields. According to [Raaijmaker, Shiffrin 81], a compound retrieval cue retrieves only items that are associated to all of its elements, but not items that may be highly associated to one item, but not to another.

One can think of an associative net as a matrix. The row and column entries of the matrix are the items of the database, in this case, books, journal articles, and authors from the HELGON literature/person database. Since we do not know the actual associative strength values between the items in that database, we must use some approximations. For this purpose, we compute proximity values between all the items in the matrix based on various aspects of their surface similarity: same name of author, same first name, common words in the title, publisher and year of publication, and so on. The resulting matrix is the "long-term memory" from which we want to retrieve.

Retrieval occurs by asking the user to set up a retrieval cue. This may be done directly in response to the given retrieval instructions, or may involve retrieval by reformulation, perhaps after some direct attempts had failed. We compute the proximity values between all items in the matrix and the components of the retrieval cue, in exactly the same way as these values were computed for the other items in the matrix. The resulting input vector is then multiplied with the long term memory matrix, yielding an output vector which provides a measure of the degree of similarity between the retrieval cue and each item in the database. The most similar items in this vector are retrieved by the system.

In principle, RETRIEVE could have all the support features for retrieval by reformulations that were built into

HELGON. HELGON users, on the other hand, could take advantage of the clear, logical hierarchy structure of the database, while no such inherent order exists in an associative net. This does not mean, however, that one cannot exploit whatever structure and organization there is in an associative net. Spreading activation processes have played a large role in memory models, from semantic nets [Loftus 73] to PDP nets [Mozer 84; Bein, Smolenksy 88]. Unlike hierarchies (or scripts, frames, schemata, and the like), the structure in associative nets is a local one. The same local relations thus can play a role in many different contexts, depending upon what else is activated. Spreading activation lets us exploit this local structure to build more efficient retrieval systems.

The problem our users usually had was that they didn't know what something was called in the database, but what they tried often was not far off. If one could take their retrieval vector, then use that as a basis for spreading activation to neighboring parts of the LTM net, perhaps something useful might turn up. We are already letting the users take advantage of the structure in the database through retrieval by reformulation: They can see what there is, and thus figure out how they should ask for what they want. But there could also be a more automatic way of taking advantage of the associative structure: Given an output vector, if we spread activation around it, database entries which are related to many of its activated items will also become activated.

We nad a task: information retrieval in a complex system that requires that the users understand it correctly, i.e., they had the right system model.

We had a problem: how to transform the naive users' situation model into an adequate system model.

We found some solutions by designing systems based on psychological principles of memory retrieval: retrieval by reformulation to support the control processes involved in finding a good retrieval cue (HELGON), and associative retrieval modeled after human memory retrieval, including spreading activation processes to exploit the structure inherent in an associative net (RETRIEVE).

We have now described two information systems that we have developed. Both share the characteristics that they attempt to help users by making it easy for them to see what the properties of the system model have to be. They differ in the nature of the knowledge base from which information is to be retrieved: In the one case, a clearly structured hierarchy (HELGON), in the other an associative network (RETRIEVE). In Section 3 of this report, we show that these are complementary approaches, not contradictory, and sketch plans for a system that combines positive features from the systems we have built so far. In Section 2.3, however, we describe another part of our project, where the goal is not merely to help the user understand what the system model is, but where we attempt directly to construct the system model for the users. That is, of course, what experienced users do when they perform routine tasks. Our system will, therefore, be a simulation of such expert performance, in the hope that we later can use what we have learned for the design of systems that would help naive users perform similar tasks.

### 2.2.4 NEWSCOPE

#### Breaking Down the Problem Space

We have divided the information management problem into four distinct processing steps. The first of these is send-time processing. Send-time processing is the work a user must do in order to send a message. This processing currently ranges from simply specifying a destination and subject (as in UNIX e-mail), to coarse grained categorization (as in USENET newsgroups), to fine grained structuring of the semantics of the message content (as in the INFORMATION LENS [Malone, Grant, Turbak 86] and OBJECT LENS [Lai, Malone 88] systems).

After send-time processing the next step is read-time processing. Read-time processing is the work a user must do in order to find and read the interesting information from within the complex information store (the complexity here may be inversely related to the amount of required send-time processing). This involves a range of activities from simply browsing the information store (like the UNIX RN news reader), to writing specific rules to filter out information which already includes sophisticated semantics (like the LENS systems).

Once a message has been read, a decision must be made as to whether or not the message should be saved for future retrieval. If the decision is for saving the information, the users then have an additional opportunity to add their own retrieval semantics to the message. We call this storage-time processing. This currently ranges from simply saving all messages to a flat file or mail box (like many people use UNIX e-mail), to adding coarse grained semantics by saving to a folder structure (like some people use UNIX e-mail), to utilizing fine grained semantics and saving to a public area which others can peruse using rules (like the LENS systems).

Finally, there is the processing which must take place when the information is needed at a later time. This is the information retrieval problem as discussed in the literature, and we call this question-time processing. Question-time processing currently ranges from simply searching for text strings, to utilizing coarse grained semantics (like the UNIX e-mail folders), to utilizing fine grained semantics in the information (like the HELGON or LENS systems).

#### New and Different Domains

In addition to the HELGON and RETRIEVE work, we have implemented the NEWSCOPE system to further investigate the issues involved with complex information stores. These issues include the structural and situation/system model ones described in Section 2.1, but much of our desire to implement NEWSCOPE was also based on the goal of exploring information stores other than the literature domain which was the current focus of the HELGON studies. This meant more than just choosing a different topic for the data in the system, it meant choosing an information store which had fundamentally different characteristics than the literature domain. Towards this end, we have implemented the NEWSCOPE system which allows users to read the messages which appear on the USENET news system.

One major difference between the literature and news domains is that *they evolve differently over time.* For instance, each literature entry which is added to the HELGON database remains there forever. While it may be modified or reclassified, it generally is never removed. In the USENET news environment messages expire like clockwork and are constantly replaced by new and different messages. The job of

NewScope is centered around the read-time task of deciding which small portion of the available information will be perused, and the storage-time task of deciding which of the perused messages are worthy of classification and storage. The difference in size between the two domains is in the orders of magnitude.

Another fundamental difference between the literature and news domains is that each user does not have to scan each new literature entry as it arrives. If any classification or translation of data format is necessary, a database administrator can take care of that and make the information available for later perusal. In other words, for the end user there is no read-time processing required to make use of a database in the literature domain. Literature is a storage-time (through modification and restorage) and question-time domain.

Finally, the literature database is controlled and updated by our own research group which makes for a nice stable environment for study, but doesn't necessarily represent the evolution of data in a public database. The hierarchy of the news system will inevitably change (it was designed with exactly that in mind), as will the the hierarchy in the literature domain (especially if the hierarchy is personalized by each user), but the difference is that in the Usenet environment these changes are out of our control. This means that users must find interesting information by browsing in a hierarchy which corresponds to somebody elses semantics (actually in the case of Usenet it's a group of "somebody elses"). So NewScope is designed to let users modify and personalize the actual newsgroup hierarchy.

### NewScope and Structure

Using the virtual newsgroup mechanism (described in Figure 2-3), a new hierarchy of newsgroups is created over time which corresponds to the semantics of the user's situation model. Also, since the filters are defined specifically to weed out uninteresting messages and include interesting ones, users are reading newsgroups which contain a higher percentage of messages they actually want to read.

In addition to redefining the newsgroup hierarchy through the creation of virtual newsgroups, NewScope displays messages in a hierarchy based upon the conversational threads of the messages. This means that the original posting of a thread constitutes the root node of a hierarchy within the containing newsgroup. The second level of the hierarchy represents all of the messages which were posted as direct responses to to this original posting. The hierarchy continues as more messages are added to the conversational thread. This allows users to read an entire thread without having to ignore messages from other threads that have arrived intertwined with the current thread. Unfortunately, most news reading programs organize messages within a group chronologically which forces the user to deal with these intertwined message threads. In addition, this hierarchical organization will allow future versions of the NewScope system (namely the InfoScope system described later) to provide operations such as saving or ignoring an entire message thread, and printing an entire thread. This means that once a thread has been selected for an operation that any future messages which arrive in that thread can be automatically included in that operation.

### NewScope and the Situation/System Models

As mentioned earlier, many of our projects are based on the theoretical framework defined by the situation versus system model distinction. Specifically, we wish to span Norman's *gulfs of execution and evaluation* [Norman 86] by bringing the system and situation models closer together. There are two basic

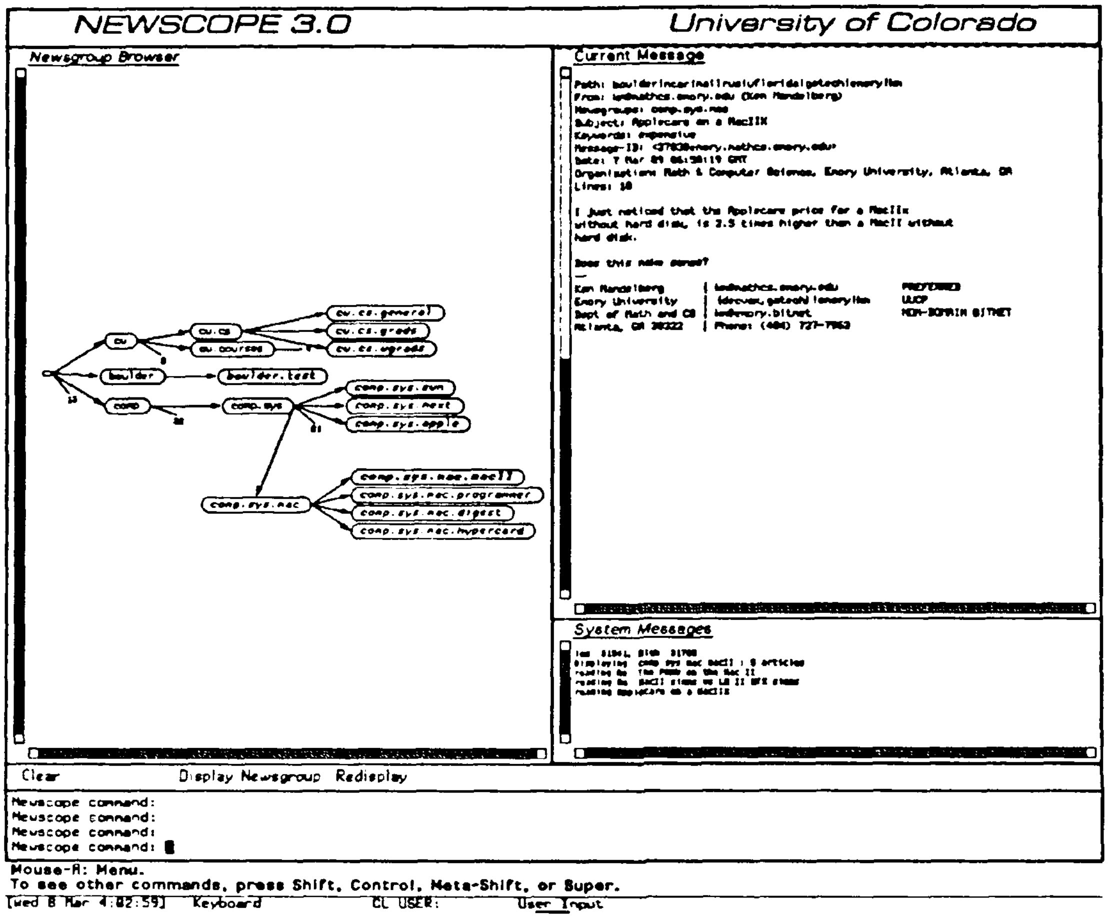


**Figure 2-3: NewScope**

NewScope is a Usenet news reader which allows users to reorganize the structure of the predefined newsgroup hierarchy (the default system model). This is accomplished through the creation of *virtual newsgroups*. They act as user defined filters which remove a certain amount of unwanted information from an existing group (Usenet or virtual) or create a new group consisting or interesting articles from a another group. For example, the newsgroup comp.sys.mac is a huge group which receives as many as a hundred new messages per day. But what the user is really interested in are articles concerning the Macintosh II computer. In order to do this he clicks the mouse on the comp.sys.mac newsgroup and chooses **create virtual newsgroup** from the resulting menu. He then has the opportunity to define a filter which will create the new newsgroup which he chooses to call comp.sys.mac.macII for convenience. He specifies that the newsgroup should contain any message posted to comp.sys.mac which contains the words macII, color, or pmmu. These are all words which identify a message as being related to the macII. Furthermore, he stipulates that the new group should not contain any messages about the SE computer. The resulting modified newsgroup hierarchy is shown in this figure.

---

directions from which this problem can be attacked. Systems such as Helgon and Retrieve are designed to move the user's situation model closer to the system model presented on the computer screen. This occurs through the process of making the data and structural organization of the information store easier to understand through visualization techniques and the presentation of examples. As queries are reformulated, they are mapped closer and closer to the system model. The NewScope system investigates attacking this problem from the opposite direction. In NewScope, users actually modify the portion of the system model which is contained in the display of the Usenet newsgroups. In effect, the users move the system model closer to their already existing situation model. While this does not eliminate the need for system model understanding on the part of the user, it does reduce it over the long run as the new hierarchy represents more and more of the user's view of the information store.

Finally, in addition to the two directions from which this system/situation model issue can be approached, it can also be attacked through the use of differing types of operations. In the HELGON and RETRIEVE systems, for example, the user is given help in deciphering the system model through reformulation, visualization, and multiple specification techniques. As this occurs, the user gains a better understanding of the system model and can therefore do a better job of mapping the situation model onto the system model. As opposed to giving help with the system model, the NETWORK system [Mannes 89] translates the user's natural language, situation model-based query into the system model for the user. This effectively eliminates the need for the user to fully understand the system model. To round out our research into the various methodologies for dealing with the situation model to system model mapping problem, the NEWSCOPE system allows the users to build their own system model which closely corresponds to the users' own situation model over time. We feel that with further investigation we will be able to properly leverage each of these methods and combine them into an integrated system for information management.

## 2.3 Knowledge Retrieval in Routine Computing Tasks[2]

In this project, we focussed not on retrieval of information from an external source, but on knowledge retrieval in routine computing tasks. We all perform such tasks easily and without engaging in much deliberate problem solving. We just "understand" what we are supposed to do. On closer analysis, however, it becomes obvious that a great deal of knowledge is needed to perform these tasks. How is it that just the relevant knowledge is activated (in the order it is needed), and not the myriad of irrelevant things we know? An answer to this question is not just of psychological interest, but has important implications for system design. At present, the only system we have built related to this part of the project is a computer simulation of human performance on these tasks [Mannes 89], though many of the insights gained from this simulation have also been incorporated into other system building efforts.

### 2.3.1 Theoretical Background

The work described here is a direct outgrowth of related work in the area of discourse comprehension and problem solving [Dijk, Kintsch 83; Kintsch, Greeno 85; Kintsch 89]. Specifically, it combines two lines of research: the work on mathematical word problems and the construction-integration model. With the word problem research, it shares the conceptualization of the problem in terms of situation versus system model: The user's task is well defined in situation model terms, and the problem is to express it in system model terms. The construction-integration model provides the mechanism for that transformation.

The construction-integration model [Kintsch 89] assumes that in discourse comprehension knowledge associated with the text is being activated in parallel and in a context-independent manner. The text plus the activated knowledge, however, form an interrelated network (with both facilitatory and inhibitory links), so that it is possible for a relaxation mechanism to discard those knowledge elements that were activated

[2]This section is a summary of [Mannes 89].

but do not fit into the discourse context. Thus, understanding is conceptualized not so much as a problem solving procedure, but akin to perception; it is basically data-driven and bottom-up, rather than expectation-driven and predictive. Context effects arise because of the integration phase which follows the network construction.

### 2.3.2 A Simulation of Human Performance

The empirical basis for the simulation was provided by a verbal protocol study with subjects performing routine computing tasks such as "You are reading your e-mail and receive a request from someone to send him a specific paragraph from a certain paper you are writing," or "Include an address you know in a letter that already exists, and then send it to someone." Although subjects had perhaps never done such a task in precisely the way it was requested here, they were thoroughly familiar with tasks of this nature and able to perform them routinely.

#### The Knowledge Base

The verbal protocols were propositionalized to obtain one part of the knowledge base for performing these tasks. The other part consisted of plan elements. A plan element is a unit of knowledge (a proposition) about how the computer system works. By selecting and sequencing a series of plan elements, plans for performing the requested actions are formed. Forming such a plan is the main feature of understanding the routine tasks studied here. A plan element consists of a name (such as "Reply-to-message-by sending-file," conditions in the world, which must exist to permit the plan element to fire (such as "Know-where-File-is," "Reading-message"), and consequences in the world of the execution of a plan element (e.g., "File-received").

A network is constructed by connecting all propositions (including plans) that share arguments. Among the set of plan elements, a different algorithm is used to determine causal connections: Each plan element activates all other elements that create as their consequence a condition it needs; and each plan element inhibits all plan elements that destroy as their consequence a condition it needs.

#### The Construction of a Net for a Specific Task

Given some task instruction, the simulation constructs a net that contains the following parts. The propositions corresponding to the text of the instruction, some of which are marked as states of the world (e.g., "There exists a file called Letter"); other propositions which were associatively activated from the knowledge base; and the set of all plan elements.

The connections among these elements are determined in the same way as in the knowledge base, plus a few additional principles. Requests in the task instruction are directly connected to corresponding plan elements (e.g., the request to "Send a message" is connected to all plan elements that result in this action). Secondly, states of the world and plan elements are connected in such a way that plan elements which have as a consequence a state of the world that already exists are inhibited.

## Planning as Integration

A vector representing all the elements in the net thus constructed is multiplied with the coherence matrix which was obtained as a result of the previous step. When the pattern of activation stabilizes, the system tries to fire that plan element that is most strongly activated. If the conditions for that element are present in the world, it fires. If not, the next most highly activated element is tried.

Once an element fires it creates a new state of the world, resulting in a slightly changed network. The relaxation process is now repeated with the new net. This cycle continues, until a plan element fires that produces the desired outcome.

Understanding instructions for routine computing task in this model implies coming up with a sequence of plan elements which, when executed in sequence, will have the desired result. Thus, for the "Send" example mentioned above, the system will first leave the mail system, then look for the file containing the paper with the target paragraph, enter the editor, copy the paragraph to a new file, leave the editor, enter the mail system, and reply to the original message by sending the file it had created. A plan for an action that turns out to be reasonably complex has been formed, not through the conventional procedures of problem solving theories (creating subgoals, searching problem spaces, etc.), but through the same comprehension mechanisms that give an account for story understanding, learning from text, etc. We claim that this may be a psychologically more accurate description of what people actually do in these situations, and that people resort to deliberate problem solving only when these highly automated, unconscious comprehension strategies fail.

A preliminary description of this research is given in [Kintsch, Mannes 87], and a full description may be found in the PhD dissertation of Mannes [Mannes 89].

## Design Implications

It is obvious that if we had a system that could comprehend a large variety of tasks such as the ones described above, this would make an impressive help system. Naive users could just insert verbal task descriptions, and the system itself would work out the appropriate system model they would need to perform these tasks. For that to be possible, we would need not a fragile prototype, but a robust large-scale system. We are far from that goal, but we outline below (Section 3.3) how our simulation could be used for the design of somewhat less utopian help systems.

We comment on one further implication of our simulation results for the design of user-centered systems, because it will be of importance for our research plans for the remainder of our project period. One could characterize the Mannes work as follows: Out of an associative knowledge base that lacks global structures such as schemata or scripts, task-specific global structures are created - the plans as a system model. Structure is not fixed, but can be generated from only locally organized associative nets upon demand (see also [Kintsch, Mannes 87]). If so, this is a crucial point for design, to which we turn below.

## 3. Plans for the Future

We propose to work along lines quite compatible with our plans for Years 3, 4, and 5 of the project as outlined in our original proposal. Nevertheless, we have learned a lot since that proposal was written, and a more focussed and precise statement of our goals for the next two years becomes possible: a broadening and shift of emphasis in our theoretical work, a redirection of the system building efforts, building upon what has already been achieved in that area, and a greater emphasis on experimental work, which now becomes possible, because theory and systems have been pushed to a point where they provide a framework for meaningful experimentation.

### 3.1 Further Elaboration and Instantiation of the Guiding Principles

The distinction between situation model and system model, which was made by analogy with other research areas, has proven to be extremely useful in our work and will continue to be important. What we have done so far, however, has also helped us to widen our perspective, and to identify even broader, and hopefully equally promising, research questions. In our work with situation and system models, we have more and more come to the point where we regard the problem of structure as a central and basic one. Situation and system models are mental representations of structures; situations and systems are structures in the world. How is one related to the other, how are mental structures represented, how are they acquired? The future directions of research to reduce the gap between the situation and system model are illustrated in Figure 3-1.

We are placing *the concept of structure* into the theoretical center of our work. We have much to go on: We have studied how structure helps performance, but we have also seen what the costs associated with structure are (e.g., in HELGON and INFOSCOPE). We have built a large scale simulation of how structure can be generated on demand in the context of particular tasks from a flexible, unstructured (except for local links) knowledge base (the Mannes work). There are clear design implications here: Structure can be helpful if it is the right structure; but systems must be constructed in such a way that they are readily reorganized in response to changing task demands. This is a very different design philosophy than one based on stable scripts and schemata.

Our theoretical work will continue to focus on this central question: how to generate structure from a flexible, associative knowledge base in the context of specific task demands. We have shown that this is possible, but many questions remain unanswered. A serious roadblock in that respect turns out to be a methodological one: how to determine empirically what a mental structure actually is like. As described in Section 2.3, it is possible to use verbal protocol methods, but such protocols provide rather indirect pictures of mental structures. Hence more direct, experimental methods for determining mental structures will be explored. We have begun to work with restricted association and clustering tasks, as well as appropriate statistical methods, to explore the possibilities for creating associative nets that more accurately represent user knowledge. Pilot results, so far, are encouraging.

Thus, we are planning studies in which "maps" of the knowledge naive and experienced users have of a system (e.g., knowledge of spreadsheets, like LOTUS 123) are created. These structures will then be used as the basis for both empirical work (learning and performance studies) and theoretical work (predicting and/or simulating the empirical studies). The purpose of these studies is to provide an empirical basis for the principle of generating structure from an associative knowledge base. The system

| | | |
|---|---|---|
| (1) | Situation Model                    System Model | no support system |
| (2) | Situation Model  <------ System Model | NewScope/InfoScope |
| (3) | ------> Situation Model  System Model | Helgon |
| (4) | Situation Model  ==agent==> System Model | Network |
| (5) | Situation Model  =  System Model | Training |

**Figure 3-1:** Different Approaches to Bridging the Gap Between the Situation and System Model

**(1) – The "normal" situation:** People have difficulties in solving problems or finding information, because there exists a gap between the situation model and the system model.

**(2) – The NewScope/InfoScope approach:** A new system model is constructed which is closer to an individual's situation model.

**(3) – The Helgon approach:** By making the system model transparent, users can bring their situation model closer to the system model. The query starts in the situation model and is incrementally reformulated into the system model.

**(4) – The Network approach:** This is the agent approach where a knowledge-based assistant translates a request in the situation model into the system model.

**(5) – The training approach:** Users are trained to express themselves in the system model.

building efforts described below are based on the same idea that structure can be very useful, but must be flexible, reorganizable -- and hence possible to generate from smaller building blocks in the context of particular tasks.

## 3.2 Complex Information Stores

On the one hand, there are a number of significant improvements which can be made within the framework of the existing Helgon system. On the other hand, we plan to construct a successor to Helgon which incorporates the most important features of the systems we have built so far and which instantiates the theoretical principles we have developed so far.

### 3.2.1 Extensions to Helgon

There are a number of improvements and extensions in the current version of Helgon which can be made, concerning non-random example selection, incorporation of a user model, and further support for editing and browsing.

HELGON also has the potential to become a meta-tool and serve as a *shell* for a broad class of information management tasks. HELGON, or aspects thereof, will be incorporated into other systems currently under development, such as INFOSCOPE (see Section 3.2.2) and CRACK and FRAMER (see Appendix III).

### 3.2.2 An Integrated Information Management System for New Domains

In our past research efforts, HELGON has been applied to well-understood problem domains, e.g., literature, persons, and electronic mail. An important characteristic of these domains is that many users are familiar with the basic abstractions used to structure these domains. In our future efforts, we will investigate new domains with the following characteristics:

- information stores where the abstractions are not well established and not shared across user communities (e.g., modern user interfaces for high-performance personal work stations),
- evolutionary information stores which grow dynamically over time and where the structure to organize them is not given at the beginning but evolves as the information store grows (e.g., slides, file directories, personal information environments).

We envision an information management system with HELGON as its interface and a parallel retrieval mechanism as in RETRIEVE (including the spreading activation feature). The database will be associative, but with the possibility of superimposing other kinds of structures (e.g., hierarchical), if desired for certain purposes (NEWSCOPE). We will explore experimentally how people choose to organize such information stores, giving them a variety of means for the establishment of associative links (content overlap, temporal contiguity, co-retrieval, etc.) and other types of structure (as in NEWSCOPE). We will also explore the possibility of giving the system some semantic knowledge by connecting it with WORDNET (Miller, presentation at 1989 ARI contractors' meeting), but have not yet determined whether these systems are compatible. Finally, computational considerations will make it necessary to explore the possibility of using a connection machine (available at our university) to implement the actual retrieval process. Discussions with Thinking Machines Corporation are under way about the use of software developed for this purpose (their software is built along the lines of our RETRIEVE system).

#### From NEWSCOPE to INFOSCOPE

Another aspect of our future research involves the actual development of INFOSCOPE from the skeleton provided by the NEWSCOPE system. The NEWSCOPE system is a news reader only. Users can browse newsgroups, read messages, and create their own newsgroups through the virtual newsgroup mechanism. Users cannot, however, save messages, respond to messages through e-mail, post follow-up messages, or define virtual newsgroups which inherit from more than one parent newsgroup. While these restrictions are severe, we have learned much through the system's development and use. Virtual newsgroups have been created that contain nearly 100% interesting information in them and therefore make the news reading experience much more efficient and enjoyable.

The move from NEWSCOPE to INFOSCOPE will first attack the deficiencies mentioned above. The ability to inherit messages in a multiple inheritance scheme will be especially important. This will allow us to develop user agents which can autonomously or semi-autonomously create newsgroups which span the entire newsgroup hierarchy. In addition, INFOSCOPE will also allow users to read their electronic mail in the same environment in which they read their news. In fact, virtual newsgroups will be possible which inherit from both newsgroups and electronic mail boxes. In other words, virtual newsgroups can be created which contain certain classes of information no matter what or where the source of that information might be in the actual newsgroup or e-mail structure.

In addition to these modifications to the read-time environment, INFOSCOPE will enhance the storage-time environment as well. Storage-time is the users' opportunity to add their own semantics which will make later retrieval easier. INFOSCOPE will have agents helping the users do this, but it will also save messages in a format that is compatible with the question-time systems we have developed (HELGON and RETRIEVE). The end result should be a fully integrated information management system which spans the information process from read-time to question-time.

### Why Build INFOSCOPE?

A major goal of our research is reducing the complexity in information stores, particularly evolving ones like USENET news, through various means. Another approach to this problem was developed in the INFORMATION LENS [Malone, Grant, Turbak 86] and OBJECT LENS [Lai, Malone 88] systems. In these messaging systems, the user is forced to add categorization and retrieval semantics at the time the message is sent. This allows users to write rules which retrieve appropriate or important messages.

We have chosen to explore a slightly different mechanism for adding semantics to messages which does not require the sender of messages to expend cognitive energy to structure the message like the LENS systems. We will address the difficult problem of creating an information system which can operate without predetermined or forced structure on the sender or recipient of electronic information. This is carried out by allowing users to define their own structure at read-time and storage-time.

For a more in depth examination of INFOSCOPE see [Fischer et al. 89] and *INFOSCOPE: Comments, Clarifications, and Plans* (a working paper distributed to M. Drillings, ARI).

## 3.3 Creating System Models

The work of Mannes [Mannes 89] has shown that it is possible for a system to understand a natural language input describing a routine computing task, and to create a system model -- a plan -- for the task at hand. Incorporated in a large computer system, such a simulation would provide a powerful help system. However, to build such a system would be quite beyond the boundaries of our project -- it would be a development project rather than a research project.

## 3.4 Conclusions

We are studying user-centered system design, specifically personalized information management systems.

Psychologically, our research question concerns the problem of structure in human knowledge. We build on a model of peoples' understanding and use of knowledge in comprehension, to investigate specific points about the role of structure in situation and system models.

This model of comprehension has been used as a basis for evaluating computer systems (as with HELGON) and to suggest principled ways in which these systems could be further developed (e.g., RETRIEVE). Thus, we will have built not only several prototype systems that might be interesting for further development and application, but also have formed a connection between cognitive principles of comprehension, knowledge use and retrieval, and system design. Equally important, we will have greatly enriched the cognitive theory that we started out with, expanded it in major ways, and directed it towards problems of system design.

# Appendix I. Additional Information

## I.1 Research Assistants

The following researchers and students have been employed (fully or partially) during the last 2 1/2 years in our ARI project:

- Computer Science:
    - Helga Nieper-Lemke
    - Curt Stevens
    - Catherine Cormack
    - Anders Morch
- Psychology:
    - Peter Foltz
    - Suzanne Mannes
    - Thea Turner

## I.2 Contributions by Students Projects

Our ARI project has provided a context for a number of research activities conducted by students (without any financial support of the ARI project) in the form of independent studies or projects in advanced courses. These activities have investigated relevant issues of our overall research effort and are briefly described below.

### Document Examiner Analysis

The purpose of this study by Emilie Young was to examine users' information retrieval process within the SYMBOLICS Document Examiner, an interface to online documentation. The subjects were sixteen SYMBOLICS users at the University of Colorado. Data pertaining to nearly 1000 interactions has been captured on a day to day basis.

### Human Assistance in Organizing Complex Information Stores

Catherine Cormack interviewed the department chairman's secretary to find out how they organize information storage and retrieval. It turned out that an important issue is the "delegation principle": When does the chairman ask his secretary to do something, when does he do it himself?

### Evaluation of ARGON

An evaluation of ARGON [Patel-Schneider, Brachman, Levesque 84], an information retrieval system that is also based on the retrieval by reformulation paradigm lead to the development of HELGON. ARGON was tested using a literature/person database and a database of electronic mail messages, and the following shortcomings were identified: It was not possible to specify queries directly without going through examples. The hierarchical organization of the database was invisible, and it was easy for users to get lost. ARGON was only an information *viewing* tool; the users could not use the same interface to *modify* the information. More details can be found in [Nieper 87].

### Retrieval by Reformulation in the Document Examiner

Jim Sullivan investigated the role of the SYMBOLICS Document Examiner in the context of the SYMBOLICS programming environment and how the application of the retrieval by reformulation paradigm could be used to enhance its utility.

### A Pattern Matcher for HELGON

The intention of this project done by Radford Walker was to make the retrieval algorithm of HELGON more intelligent by allowing non-exact matches between the users' input and database items. Examples for non-exact matches include: substring matches (which were incorporated in the current version of HELGON), synonyms, slight misspellings, and word and letter inversions.

### Fisheye Views in HELGON

Bill Turnbull implemented a fisheye view [Furnas 86] for the graphical category hierarchy display in HELGON. Fisheye views are a way to present a large information store to the users in such a way that they see the global context but still get enough detail at points of interest. In the traditional fisheye view concept, the degree of detail mainly depends on the distance from the current point of interest, i.e., it is independent of what is important to the users. Users should therefore have the possibility to customize a fisheye view by specifying their a priori interest in certain parts of the information store.

### Input Functions in HELGON

The goal of this project was to model input functions that are available on SYMBOLICS machines in HELGON. This domain was chosen because it is one that is not as homogeneous and not as familiar to the users as, for instance, the literature domain. It turned out that the main problem was to find the right structure to represent the information, i.e., to categorize input functions in a meaningful way and to find the right attributes for the information associated with one input function.

### Zero-Matching Items Problem in HELGON

In this project, Bjoern Andresen addressed the problem that users of HELGON get no support if a query specification results in "0 matching items." They may assume that an item with the desired properties doesn't exist, but the real reason for the null response may be an incorrect specification. And even if the specification is correct, the users might want to know which restrictions could not be satisfied at the same time and eliminate or modify one of them.

## I.3 Workshops Organized by the Project

- Breckenridge Workshop 1987: "Personalized Intelligent Information Systems [Fischer, Nieper 87]"
- Breckenridge Workshop 1988: "Mental Models and User-Centered Design [Turner 88]"
- Vail HCIC[3] Workshop 1989: "Foundations of Human-Computer Interaction"
- HCIC Workshop planned for 1990: "Information Access"

[3]The Human-Computer Interaction Consortium (HCIC) was founded in 1988.

## I.4 External Collaborations

### Thomas Landauer, Bellcore, Morristown, NJ

The research group at Bellcore under the direction of Tom Landauer has investigated a number of interesting problems which are directly relevant for our research project (e.g., the vocabulary problem [Furnas et al. 87] and fisheye views [Furnas 86]). Peter Foltz, who has worked on the ARI project in the past, is currently doing an internship at Bellcore and will return to CU later. Peter's stay will provide us with a unique opportunity to intensify our research collaboration.

### Hans Brunner, U S WEST, Denver, CO

Hans Brunner, Scott Wolff, and Andy Parng have been working on various systems in the Intelligent Customer Assistance project. In particular, Andy has implemented the IDEAS system, a query system based upon the retrieval by reformulation paradigm. IDEAS extends the work we have done on the HELGON system by adding a new query specification tool. This tool allows users to specify parts of the query in a graphical manner. For example, to find all houses for sale in a two mile radius of a certain point, the user specifies that the subject of the initial query is homes for sale. He then draws a two mile circle on a map provided by the system. The display then indicates where any candidate homes are by flashing them on the map. At this point the user can zoom in on the map to make a more specific graphical query, or can reformulate the other part of the query. This might include a specification of the acceptable price range for the home search in question. In this way, there is very little effort necessary on the part of the users in order to evolve the specification from their situation model to the system model (assuming that the users know how to read a map).

### Mike Atwoods, NYNEX, White Plains, NY

Gerhard Fischer visited NYNEX in the Fall of 1988 to present the research work in personal information environments and specifically to discuss the HELGON system with the research group at NYNEX. NYNEX (as other companies) faces a number of problems where the approach taken with the HELGON system offers interesting ways to tackle them. Their large information stores consist, for example, of millions of COBOL programs which were written over the last 20 years. There is nobody around who understands these information stores any more. They are heterogeneous and lack a good conceptual structure. Traditional database approaches have badly failed in tackling the problem of maintaining and updating these information stores.

### Ron Brachman, AT&T Bell Laboratories, Murray Hill, NJ

Gerhard Fischer visited AT&T Bell Laboratories in the Fall of 1988 to present the research work in personal information environments and specifically to discuss the HELGON system. Several people within this group (e.g., Brachman and Patel-Schneider) were major contributors to the ARGON system which we got from them several years ago and which served as the starting point for the HELGON system. The problems which they encounter are very similar to the problems described above for NYNEX. They are working on several new systems (e.g., a successor to the KANDOR knowledge representation formalism and a new version of an ARGON-like system) which are of direct relevance to the efforts in our project. They obtained from us (in return for giving us the original version of the ARGON system): the HELGON system, the HELGON video tape, and a new version of ARGON (converted by us from Release 6 to Release 7 on the SYMBOLICS machines).

### Erich Neuhold, GMD-F4, Darmstadt, Federal Republic of Germany

The research group in Darmstadt works in the general area of "integrated publication and information systems." In the context of a cooperation agreement between the Computer Science Department and the Institute of Cognitive Science at the University of Colorado, Boulder and them, the topics of our ARI project have played an important role. We provided them with a copy of the HELGON system and the HELGON video tape. Plans for future collaboration efforts will be worked out in the next few months.

### William Moninger, NOAA, Boulder, CO

William Moninger has been working on the METALOG system, a personal intelligent information system for the management of scientific "metadata," that is, data about scientific data. The initial system is being developed for use by the radar and lidar program areas of the Wave Propagation Laboratory, to be used in conjunction with their new data analysis workstation. The system should be applicable, however, to any scientific research that involves detailed study of large amounts of data displayed on a computer.

## I.5 Publication Record

### I.5.1 Publications in Journals and Conference Proceedings

1. G. Fischer: **Cognitive View of Reuse and Redesign**, IEEE Software, Special Issue on Reusability, July 1987, pp. 60-72.

   The microelectronics revolution of the 1970s made computer systems cheaper and more compact – and greatly increased the range of their capabilities. Much of the available computing power is wasted, however, if users don't understand and use the systems' full potential. In the past, too much attention has been given to systems technology and not enough to the effects of that technology – which has produced inadequate solutions to real-world problems, imposed unnecessary constraints on users, and failed to respond to changing needs.

2. G. Fischer: **Making Computers more Useful and more Usable**, Proceedings of the 2nd International Conference on Human-Computer Interaction (Honolulu, Hawaii), Elsevier Science Publishers, New York, August 1987, pp. 97-104.

   Useful computers which are not usable are of little help; but so are usable computers which are not useful. One of the major goals of our research is to achieve these two goals simultaneously. We claim that useful computers have to be complex systems with a rich functionality. To make these systems usable and to exploit their power, we have constructed a variety of computer-based intelligent support systems which take advantage of the interactive and dynamic nature of computer systems. Information and knowledge embedded in computers can be represented and used in qualitatively different ways than on paper. Paper is passive and can only serve as a repository for information, whereas computer systems can be active and assist us in searching, understanding, and creating knowledge in the course of cooperative problem solving processes.

3. G. Fischer, C. Stevens: **Volunteering Information – Enhancing the Communication Capabilities of Knowledge-Based Systems**, Proceedings of INTERACT'87, 2nd IFIP Conference on Human-Computer Interaction (Stuttgart, FRG), B. Shackel (ed.), North-Holland, Amsterdam, September 1987, pp. 965-971.

   Cooperative problem solving systems support the solution of tasks which cannot be solved by the human or the computer alone. These systems need to be knowledge-based and require flexible communication paradigms allowing natural communication with both experts and novice users of the system. Natural communication (quite different from natural language) has to support mixed-initiative dialogues where information can be volunteered by the system and the user. In this paper, we present prototypical systems which assist users in rebooting a computer. REEBOTER is a rule-based system which guides the user with a strongly system-directed dialogue through this task. The use of this system has shown that the communication paradigm was too narrow to make it a worthwhile tool (especially for the expert user). The SYSTEMS ASSISTANT tries to overcome the noted shortcomings by allowing the users to interact with the system in a mixed-initiative dialogue, to volunteer information and to deviate from the system generated discourse structure.

4. G. Fischer, A.C. Lemke: **Construction Kits and Design Environments: Steps Toward Human Problem-Domain Communication**, Human-Computer Interaction, Vol. 3, No. 3, 1988, pp. 179-222.

Our goal is to build cooperative computer systems to augment human intelligence. In these systems the communication between the user and the computer plays a crucial role. To provide the user with the appropriate level of control and a better understanding, we have to replace human-computer communication with *human problem-domain communication*, which allows users to concentrate on the problems of their domain and to ignore the fact that they are using a computer tool. Construction kits and design environments are tools that represent steps towards human problem-domain communication. A construction kit is a set of building blocks that models a problem domain. The building blocks define a design space (the set of all possible designs that can be created by combining these blocks). Design environments go beyond construction kits in that they bring to bear general knowledge about design (e.g., which meaningful artifacts can be constructed, how and which blocks can be combined with each other) that is useful for the designer. Prototypical examples of these systems (especially in the area of user interface design) are described in detail and the feasibility of this approach is evaluated.

5. G. Fischer, C. Rathke: **Knowledge-Based Spreadsheet Systems**, Proceedings of AAAI-88, Seventh National Conference on Artificial Intelligence (St. Paul, MN), Morgan Kaufmann Publishers, San Mateo, CA, August 1988, pp. 802-807.

Spreadsheet systems have changed the way the world perceives and deals with computers. In an attempt at maintaining the positive elements of spreadsheets while overcoming some of their limitations, we have developed FINANZ, a computational environment to develop financial planning systems. FINANZ contains a form-based user interface construction system, which allows the creation of advanced user interfaces without the need for conventional programming. It uses constraint-based programming for the representation of knowledge about the application domain. Its layered architecture (based on object-oriented knowledge representation) supports the modification and extension of the system and the dynamic generation of explanations.

6. G. Fischer: **Cooperative Problem Solving Systems**, invited contribution to the 1st International Symposium on Artificial Intelligence (Monterrey, Mexico), October 1988, pp 127-132

Problem solving effectiveness is often enhanced by cooperation -- traditionally, cooperation among people, or, more recently, cooperation between a human and a computer. The emphasis of our work is on creating computer systems to facilitate the cooperation between a human and a computer. Examination of these systems provides evidence that learning and effective problem solving can be improved through the use of cooperative systems. It also indicates the need for a richer theory of problem solving, which would analyze the functions of shared representations, mixed-initiative dialogues, and management of trouble, if problems occur.

7. G. Fischer: **Human-Computer Interaction Software: Lessons Learned, Challenges Ahead**, IEEE Software, Vol. 6, No. 1, January 1989, pp. 44-52.

Human-computer interaction (HCI) is an ill-structured domain that is limited much more by specification than by implementation. When we write software for HCI, we define not only what the computer will do but also what humans will and can do – and we make assumptions about what they want to do. Humans are individuals.

They have different knowledge and different preferences, and they change; that is, a novice may become an expert over time. In our research over the last ten years, we have tried to improve human-computer interaction by widening the explicit communication channel between humans and computers and by establishing with knowledge-based systems an implicit communication channel in which shared knowledge is the basis for cooperative problem solving. Our efforts have taught us a number of lessons such as the limitations of user interface toolkits, the need for higher level, application-oriented abstractions, and the need for intelligent support systems. We have defined several challenges for the future: the need for methodologies and tools for coping with design tasks with incomplete specifications, the challenge of resolving design trade-offs, and the need to support cooperative problem solving.

8. G. Fischer, R. McCall, A. Morch: **Design Environments for Constructive and Argumentative Design**, Human Factors in Computing Systems, CHI'89 Conference Proceedings (Austin, TX), ACM, New York, April 1989.

   Design Environments are computer systems which support design by enabling cooperative problem solving between designer and computer. There are two complementary problem solving activities in design: constructive design and argumentative design. We have created two computer-supported environments, CRACK and VIEWPOINTS, to support these two activities. CRACK is a knowledge-based critic which has knowledge about how kitchen appliances can be assembled into functional kitchens. VIEWPOINTS is a hypertext system based on the IBIS design methodology and contains useful information about the principles of kitchen design. The integration of these two types of systems will eliminate shortcomings of the individual systems.

9. G. Fischer, H. Nieper-Lemke: **HELGON: Extending the Retrieval by Reformulation Paradigm**, Human Factors in Computing Systems, CHI'89 Conference Proceedings (Austin, TX), ACM, New York, April 1989.

   People who attempt to use a complex information store on a computer encounter a number of problems: They do not know what information exists or how to find information, they get no support in articulating a question, and they are unable to phrase their question in terms that the system understands. HELGON, an intelligent environment that supports limited cooperative problem solving, helps people deal with complex information stores. HELGON supports retrieval and editing by reformulation with multiple specification techniques, and it acquaints the user with the system model of the information store. Within the current HELGON system, a number of different information stores have been implemented. Empirical evaluations have shown that HELGON supports effective communication. In addition, the evaluations have shown interesting extensions for future work.

### I.5.2 Technical Reports

1. G. Fischer, H. Nieper (eds.), **Personalized Intelligent Information Systems, Workshop Report (Breckenridge, CO)**, Institute of Cognitive Science, University of Colorado, Boulder, CO, Technical Report, No. 87-9, 1987.

2. H. Nieper: **Information Retrieval by Reformulation: From ARGON to HELGON**, in G. Fischer, H. Nieper (eds.), Personalized Intelligent Information Systems, Workshop Report (Breckenridge, CO), Institute of Cognitive Science, University of Colorado, Boulder, CO, Technical Report, No. 87-9, 1987, Ch. 19.

3. W. Kintsch: **Knowledge Assessment and Knowledge Organization**, in G. Fischer, H. Nieper (eds.), Personalized Intelligent Information Systems, Workshop Report (Breckenridge, CO), Institute of Cognitive Science, University of Colorado, Boulder, CO, Technical Report, No. 87-9, 1987, Ch. 10.

4. S. Mannes: **Modeling the Generation of Knowledge Structures: The Basics**, in G. Fischer, H. Nieper (eds.), Personalized Intelligent Information Systems, Workshop Report (Breckenridge, CO), Institute of Cognitive Science, University of Colorado, Boulder, CO, Technical Report, No. 87-9, 1987, Ch. 11.

5. G. Fischer: **Intelligent Support Systems for Hyperknowledge**, Technical Report, Department of Computer Science, University of Colorado, Boulder, CO, November 1987.

Information and knowledge embedded in computers can be represented and used in qualitatively different ways than on paper. Paper is passive and can only serve as a repository for information, whereas computer systems can be active and assist us in searching, understanding, and creating knowledge in the course of cooperative problem solving processes. We have explored this general idea (characterizing some of the main aspects of hypertext systems) in the context of building a variety of intelligent support systems for high-functionality computer systems emphasizing the following specific issues: representation of programs as knowledge networks where the code, the documentation, and visual representations are external representations generated from the same complex internal knowledge structure; user-definable filters to give users control which parts they would like to see; constraint mechanisms to maintain consistency between internal and external representations; different browsing systems to explore hyperknowledge spaces and design kits as prototypes of hyperknowledge assistants. To increase the usefulness and usability of high-functionality computer systems, they should be constructed as hyperknowledge systems where the intelligent support systems are an integral part of the overall design.

6. G. Fischer: **Learning on Demand: Ways to Master Systems Incrementally**, Technical Report, Department of Computer Science, University of Colorado, Boulder, CO, 1987.

Obsolescence of knowledge and information overload are some of the major problems of our information-rich society. High-functionality computer systems (HFCS) are a prime example where users are confronted with these problems. No user of such a system will have a complete knowledge of it; these systems have to be learned incrementally. Structuring these systems as increasingly complex microworlds allows to get users started and to confront them with challenging but attainable goals. Tutoring systems fall short to support a long-term, user-driven learning process. Learning on Demand has to be supported to increase the willingness of users to overcome the low usability factor of these systems. Several systems which achieve some of these objectives will be described as instances of the general class of intelligent support systems: information retrieval systems, help systems, critics and construction and design kits. The paper concludes by describing the experiences gained with these systems and the lessons learned from them for future research problems.

7. G. Fischer: **Mental Models – A Computer Scientist's Point of View**, in A.A. Turner (ed.), "Mental Models and User-Centered Design," Workshop Report (Breckenridge, CO), Technical Report No. 88-9, Institute of Cognitive Science, University of Colorado, Boulder, CO, 1988, pp. 15-26.

In our research, we are interested in the design and understanding of high-functionality computer systems (HFCS). These systems contain thousands of objects (e.g., functions, classes, operations), and they are not completely mastered by users. In order to enhance and/or reduce the learning process of HFCS and to allow users to achieve their goals with them, we have constructed a number of intelligent support systems. These support systems should enhance the acquisition of a MM, and they should reduce the amount of knowledge which needs to be accumulated in the MM to operate the systems effectively.

8. P.W. Foltz, W. Kintsch: **An Empirical Evaluation of Retrieval by Reformulation on HELGON**, in A.A. Turner (ed.), "Mental Models and User-Centered Design," Workshop Report (Breckenridge, CO), Technical Report No. 88-9, Institute of Cognitive Science, University of Colorado, Boulder, CO, 1988, pp. 9-14.

Retrieval by reformulation is a method of extracting information from a database in which a user interactively refines partial descriptions of a target item by criticizing successive examples. Retrieval by reformulation is based on psychological theory of human memory in which people retrieve information from human memory through an interactive process of constructing partial descriptions of what they want to retrieve. We evaluated the HELGON retrieval system by recording protocols of subjects using the system and recording the errors made. The errors were classified into two groups, procedure misunderstanding errors and database misunderstanding errors. Analysis of the errors showed that a large proportion of the database misunderstanding errors were due to confusion about the use of the database hierarchy This suggests that subjects may be using associative retrieval cues to perform retrieval tasks. A possible solution to this problem is to design a retrieval system that works in the same way as human retrieval.

9. G. Fischer, A.C. Lemke: **Knowledge-Based Design Environments for User Interface Design**, Technical Report, Department of Computer Science, University of Colorado, Boulder, CO, September 1988.

   Knowledge-based systems and new paradigms for communication between software engineers and these systems are promising approaches to augmenting the designers' capabilities in the design, understanding, and modification of complex software systems. High-functionality computer systems, containing a rich base of domain-dependent and domain-independent abstractions, are a starting point for working towards this goal. They contain many building blocks and tools, but they offer no guidance on how to use them. Knowledge-based design environments, containing critics to praise good solutions and to suggest improvements to inferior designs, offer explanations and remedies that help designers overcome the bewildering set of choices inherent in design. This paper describes FRAMER, a construction kit and design environment for the design of window-based user interfaces. FRAMER supports a direct-manipulation interaction style for the development of user interfaces. It provides not only building blocks for this task but also a design environment with knowledge about the quality of a design. It praises the user for good designs and suggests improvements to inferior designs. It offers explanations for its "opinion," and in specific cases it is able to provide remedies for shortcomings. It generates program code automatically and maintains consistency between the different representations. It supports different design strategies: design by composition of domain-oriented building blocks as well as modification of complete examples. FRAMER has proven to be a powerful tool that has increased programmer productivity and contributed to the development of better products. We believe that knowledge-based design environments of this kind will play an increasingly important role in software development.

10. G. Fischer, A.C. Lemke: **FRAMER: Integrating Working and Learning**, Technical Report, Department of Computer Science, University of Colorado, Boulder, CO, December 1988.

    High-functionality computer systems pose a number of challenging problems. No user of such a system will have complete knowledge of it. Intelligent support systems that at once enhance learning and reduce the need for learning have become necessities rather than luxuries for users of these systems. In this paper, we discuss the need to integrate working and learning, describe requirements for systems that facilitate this integrated activity, and present FRAMER, a prototypical example of such systems. FRAMER is a domain-oriented construction kit and design environment that provides unobtrusive advice, praise, and criticism for users who are involved in their own activities pertaining to the design of window-based user interfaces. In addition to providing building blocks for this task, FRAMER is a design environment with knowledge about the quality of a design. FRAMER supports learning by praising good designs and explaining shortcomings of inferior designs. FRAMER provides exemplary remedies for most shortcomings it detects.

11. G. Fischer, P.W. Foltz, W. Kintsch, H. Nieper-Lemke, C. Stevens: **Personal Information Systems and Models of Human Memory**, Human-Computer Interaction, 1989, to be submitted.

12. P.W. Foltz, W. Kintsch: **Information Retrieval and Human Memory**, Journal of Memory and Language, 1989, to be submitted.

## Appendix II. Relevance for ARI – by Thomas W. Mastaglio

**Remark:** We have asked Thomas Mastaglio (who is currently a PhD student working with Gerhard Fischer and Walter Kintsch at CU Boulder) to write the following brief assessment of our research effort.

### "So What Good Is this Stuff Anyway: An Attempt to Situate some Basic Research In HCI"

By

Thomas W. Mastaglio, Lieutenant Colonel, U.S. Army

This paper is a brief attempt to bridge the gap between the needs of the Army and basic research into methods for applying artificial intelligence to the problems of human-computer interaction. The specific research I will discuss is being conducted under the sponsorship of ARI at the University of Colorado and directed by Gerhard Fischer and Walter Kintsch.

I will attempt to identify where I see potential applications for the specific issues that are being pursued in this work, that is, how to potentially apply the results of this type of research to solving Army problems. I do not claim expertise in all the application areas I cite, instead my observations are born of a general understanding based on my previous military service. Any misconceptions or overgeneralization are probably more the result of attempting to extrapolate that experience to a specific problem or opportunity. Readers having specific experience with any of these applications are encouraged to critique, modify, or expand on any of the ideas.

I will further attempt to map the research agenda into some specific needs or opportunities. That agenda addresses the problem of developing personalized user-centered computer systems, specifically to build personalized information systems. I would encourage the reader to view my specific discussion as an opportunity to get a flavor of how successes in this areas might generalize to fulfill present and future needs of the Army in a more general sense.

The work conducted in Fischer and Kintsch's ARI project attempts to address the problem of how to provide better methods to access complex computer-based information stores by designing and testing systems that integrate new ideas; concurrently, an understanding of the theoretical and methodological issues from a Cognitive Science perspective will be developed. A theory of design and comprehension for complex user-centered systems would be useful in the development of a wide variety of systems in current use in the Army, as well as a variety of future applications. Complex information stores can be found in two areas within the Army: in our tactical data systems, and in our administrative systems.

Access to tactical data is crucial for intelligence gathering, operations planning, and decision-making. The users of tactical systems tend to be non-computer trained commanders and unit staffs. These users need to be able to cull from an available data set the specific pieces of information that they need to do battlefield planning and decision making. Even as a group these users tend to have differing perspectives on the same data because they come from different professional backgrounds and have different needs. The application of findings on designing user-centered systems could find direct application in the interface and access mechanisms for such systems as Maneuver Control System (MCS), Advanced Field Artillery Target Acquisition and Data System (AFATADS), and their future replacements.

The users of administrative systems could also benefit from a more user-centered approach, examples are the current SIDPERS, JUMPS and SAILS. Users of these systems are often specifically trained on using these systems. That training seeks to develop an understanding of what the system contains and how to access that information. This research could improve not only how, but how widely, these types of systems could be used. As we combine administrative databases (for example the effort to consolidate personnel and financial records), users with significantly different world views and requirements will need to access the same data set. Too often the interfaces to these large data systems have to be learned because they are so general and have such wide functionality that complexity cannot be avoided. A better alternative is to make available multiple access methods to that same information -- one of the goals of this research.

Diverse databases are also often combined to serve the needs of staff officers involved in such functions as budget forecasting, force development, or personnel management. This coalescence of different information stores is usually done manually or with ad hoc approaches. As computational resources grow to support very large databases, sets of diverse information across the entire Army will be readily available. The problem with this schema is that unless the same data is repeated for different functional needs, introducing consistency problems, we will need access methods to support multiple user views over the same data space. The theoretical foundations for such access technologies presently do not exist, and I believe that research in personalized information systems provides a context to develop this theory.

Designing new computer systems for the Army is an area where new theoretical ideas about how to provide personalized access could also find direct application. Tactical planning systems have complex underlying databases about current unit capabilities, fire support targeting lists, logistical resources, and intelligent information. Staff officers with diverse interest and requirement all need to use this data, but for different purposes. They have different views of the world. Design of new tactical support systems would benefit from an interface approach that allows personalized retrieval of data. Another similar type of system is a simulation environment, which is used for training and conceptual analysis of force structure and tactics. Different functional areas again have differing interests and foci for each simulation result.

An off shoot of this research is to provide insight into the problem of providing knowledge-based advice to users engaged in problem solving using a computerized system. One approach, cooperative problem solving, is based on the idea that a knowledge-based computer system can play a significant role in helping a user solve a complex problem. The computer as an agent in this process will need to possess a rich repertoire of interaction capabilities in order to achieve an acceptable level of human-computer communications. The ideas and theories from basic research investigated by the project in the context of user-centered computer systems should allow the further development and testing of systems based upon this intelligent agent metaphor.

It is now technologically possible, given what artificial intelligence has found out about knowledge acquisition, to capture expertise in a computer knowledge-base that can be directly used in the operational planning and control processes. This is especially attractive for those areas that are primarily data-oriented and where the expertise surrounding that data has an objective basis (e.g., logistics management). The question is how to use this knowledge -- what is the best paradigm for making it available to the functional area expert, the commander, or another staff officer. The autonomous expert system is probably not the correct approach because many functions have a necessary human dimension. I claim that military planning and operations are such functions. In these cases the computer can serve as knowledgeable assistant for the commander or staff officer.

**Summary.** I have attempted to describe how the results from a specific, focused research effort to improve human-computer interaction relates to some specific problems, needs, or opportunities for the Army. These problems, needs and opportunities are generalized based on my personal experiences and/or suppositions about what types of systems either already exist or could be developed in the future. The purpose of this exercise is not to justify this specific research agenda or focus on any particular problem but instead to give a flavor of the potential long term contributions this type of basic research can make to the Army.

## Appendix III. Related Activities

The research conducted in our ARI research project has not only led to scientific collaboration with a large number of other research organizations (see Section I.4), but it also had a major impact on other research activities conducted within our own environment (which are supported by other research grants). Equally important was the feedback and insight which we gained through these other research projects for our work in the ARI project. Some of the most important other activities are briefly described below.

### III.1 Catalogues in Reuse and Redesign

One of the most exciting possibilities for solving the software crisis is finding ways that reuse and redesign can become major methodologies in designing the software systems of the future. New programming environments have to be created which support design methodologies whose main activity is not only the generation of new independent programs but the maintenance, integration, modification, and explanation of existing ones. One crucial challenge will be to identify, find, and adapt the objects in high-functionality computing systems which users need to accomplish their tasks -- a research problem which is the focal point of our ARI project.

We have constructed two prototypical system, FRAMER and CRACK to explore these issues [Fischer 89]. FRAMER and CRACK are design environments that support different design strategies such as design with basic building blocks and redesign of prototypical examples. FRAMER supports the construction of window-based user interfaces, and CRACK supports cooperative kitchen design.

The `Catalog` in FRAMER (Figure III-1) contains a number of prototypical designs. They can be praised and critiqued, and when brought into the work area, they can be modified and used as a starting point for redesign. Prototypical solutions that can be changed and refined through redesign are important enrichments for designers and enlarge their design possibilities. After having created an interesting design, users can also store it in the `Catalog`. We are in the process of developing design specifications for using HELGON to identify the most promising example in the catalogue.

### III.2 Understanding the Nature of Cooperative Problem Solving

Over the last few, years we have constructed a number of intelligent support systems which amplify human intelligence. These systems have proven to be quite useful -- not only as research vehicles, but in the learning and working environments in which they are used. One of their major shortcomings is that they operate as "one-shot affairs" (e.g., a system gives an error message, a piece of help information, or some documentation aspect), whereas human critics and advisors engage in cooperative problem solving dialogues. These dialogues help the user articulate a question in the first place, ask for missing information, uncover misunderstandings, make assumptions, and explain them to the user.

Cooperative problem solving systems are an exciting approach to exploiting the full power of high-functionality computing environments by augmenting human intelligence rather than replacing it. Communication problems with intelligent support systems can arise from the fact that the user's actions, rather than flowing smoothly according to a fully instantiated plan, often require interpretation in terms of a variety of situational factors. There is often a mismatch between the designer's assumptions about the meaning of the user's actions and the user's actual intentions at the time of the interaction -- an assump-

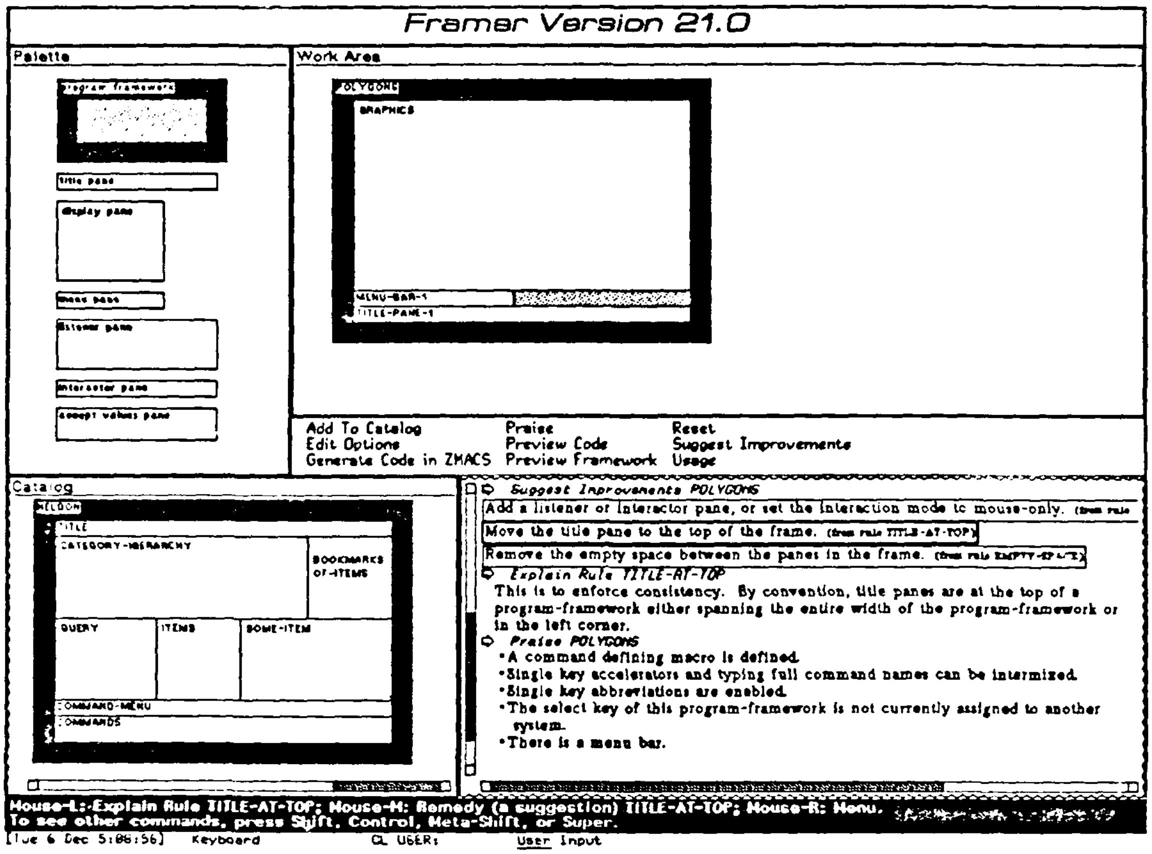


Figure III-1: FRAMER: A Design Environment for Window-Based Interfaces

tion (based on empirical observations) which provides the major design rationale for HELGON-like systems.

To deepen our understanding of cooperative problem solving, we have conducted an empirical study in a realistic setting. We chose to study salesperson-customer interactions in a large hardware store (McGuckin in Boulder). There were several reasons for this choice:

- McGuckin has a tremendous diversity of inventory so that you would expect to find more than one solution to any given problem. (At last estimate, they had over 350,000 different items, and between one and five million items in all).
- The salespeople at McGuckin have a reputation for expertise in just about every hardware domain, from plumbing to fasteners. We conducted a survey that showed that McGuckin customers expect the salespeople to be able to do more than just locate items in the store and depend on them to help solve problems.
- The store is so large that it seems impossible for one person to know it all. (They have approximately 100 salespeople for about 33,000 square feet of retail sales space).
- The store itself is a complex information space. In addition to "just" solving customer problems, the salesperson has to know where the parts are. A subtle issue here is that sometimes different departments will have slightly different items that address the same

problem from different perspectives. A good salesperson thus needs to know more than one department; they need a more global perspective.

We strongly believe that the result and insight gained through this study (which is supported by the Colorado Institute of Artificial Intelligence) will deepen our understanding of the research topics in the ARI project.

# References

[Bein, Smolenksy 88]
J. Bein, P. Smolenksy, *Application of the Interactive Activation Model to Document Retrieval*, Technical Report CU-CS-405-88, Department of Computer Science, University of Colorado, Boulder, CO, May 1988.

[Collins, Quillian 69]
A.M. Collins, M.R. Quillian, *Retrieval from Semantic Memory*, Journal of Verbal Learning and Verbal Behavior, Vol. 8, 1969, pp. 240-247.

[Dijk, Kintsch 83]
T.A. van Dijk, W. Kintsch, *Strategies of Discourse Comprehension*, Academic Press, New York, 1983.

[Fischer 89]
G. Fischer, *Human-Computer Interaction Software: Lessons Learned, Challenges Ahead*, IEEE Software, Vol. 6, No. 1, January 1989, pp. 44-52.

[Fischer et al. 89]
G. Fischer, P.W. Foltz, W. Kintsch, H. Nieper-Lemke, C. Stevens, *Personal Information Systems and Models of Human Memory*, Human-Computer Interaction, 1989, to be submitted.

[Fischer, Nieper 87]
G. Fischer, H. Nieper (eds.), *Personalized Intelligent Information Systems, Workshop Report (Breckenridge, CO)*, Institute of Cognitive Science, University of Colorado, Boulder, CO, Technical Report, No. 87-9, 1987.

[Fischer, Nieper-Lemke 89]
G. Fischer, H. Nieper-Lemke, *HELGON: Extending the Retrieval by Reformulation Paradigm*, Human Factors in Computing Systems, CHI'89 Conference Proceedings (Austin, TX), ACM, New York, April 1989.

[Foltz, Kintsch 89]
P.W. Foltz, W. Kintsch, *Information Retrieval and Human Memory*, Journal of Memory and Language, 1989, to be submitted.

[Furnas 86]
G.W. Furnas, *Generalized Fisheye Views*, Human Factors in Computing Systems, CHI'86 Conference Proceedings (Boston, MA), ACM, New York, April 1986, pp. 16-23.

[Furnas et al. 87]
G.W. Furnas, T.K. Landauer, L.M. Gomez, S.T. Dumais, *The Vocabulary Problem in Human-System Communication*, Communications of the ACM, Vol. 30, No. 11, November 1987, pp. 964-971.

[Halasz 88]
F.G. Halasz, *Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems*, Communications of the ACM, Vol. 31, No. 7, July 1988, pp. 836-852.

[Kintsch 77]
W. Kintsch, *Memory and Cognition*, John Wiley & Sons, New York, 1977.

[Kintsch 89]
W. Kintsch, *The Representation of Knowledge and the Use of Knowledge in Discourse Comprehension*, in K. Graumann, R. Dietrich (eds.), *Language Processing in Social Context*, North Holland, Amsterdam, 1989, pp. 185-209, also published as Technical Report No. 152, Institute of Cognitive Science, University of Colorado, Boulder, CO.

[Kintsch, Greeno 85]
W. Kintsch, J.G. Greeno, *Understanding and Solving Word Arithmetic Problems*, Psychological Review, Vol. 92, 1985, pp. 109-129.

[Kintsch, Mannes 87]
W. Kintsch, S.M. Mannes, *Generating Scripts from Memory*, in J. Hoffmann, E. van der Meer (eds.), *Festschrift for F. Klix*, North-Holland, Amsterdam, 1987, also published as Technical Report No. 87-3, Institute of Cognitive Science, University of Colorado, Boulder, CO.

[Lai, Malone 88]
K.-Y. Lai, T.W. Malone, *Object Lens: A "Spreadsheet" for Cooperative Work*, Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW'88), ACM, New York, September 1988, pp. 115-124.

[Loftus 73]
E.F. Loftus, *Activation of Semantic Memory*, American Journal of Psychology, Vol. 86, 1973, pp. 331-337.

[Malone, Grant, Turbak 86]
T.W. Malone, K.R. Grant, F.A. Turbak, *The Information Lens: An Intelligent System for Information Sharing in Organizations*, Human Factors in Computing Systems, CHI'86 Conference Proceedings (Boston, MA), ACM, New York, April 1986, pp. 1-8.

[Mannes 89]
S. Mannes, *Planning Routine Computing Tasks*, PhD Dissertation, University of Colorado at Boulder, Department of Psychology, 1989.

[Mozer 84]
M.C. Mozer, *Inductive Information Retrieval Using Parallel Distributed Computation*, ICS Report 8406, Institute for Cognitive Science, University of California, San Diego, La Jolla, CA, June 1984.

[Nieper 87]
H. Nieper, *Information Retrieval by Reformulation: From ARGON to HELGON*, in G. Fischer, H. Nieper (eds.), *Personalized Intelligent Information Systems, Workshop Report (Breckenridge, CO)*, Institute of Cognitive Science, University of Colorado, Boulder, CO, Technical Report No. 87-9, 1987, Ch. 19.

[Norman 86]
D.A. Norman, *Cognitive Engineering*, in D.A. Norman, S.W. Draper (eds.), *User Centered System Design, New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1986, pp. 31-62, Ch. 3.

[Patel-Schneider, Brachman, Levesque 84]
P.F. Patel-Schneider, R.J. Brachman, H.J. Levesque, *ARGON: Knowledge Representation Meets Information Retrieval*, Fairchild Technical Report 654, Schlumberger Palo Alto Research, September 1984.

[Raaijmaker, Shiffrin 81]
J.G. Raaijmaker, R.M. Shiffrin, *Search of Assocative Memory*, Psychological Review, Vol. 88, 1981, pp. 93-134.

[Schank 82]
R.C. Schank, *Dynamic Memory*, Cambridge University Press, Cambridge, MA, 1982.

[Schank, Abelson 77]
R.C. Schank, R.P. Abelson, *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

[Tou et al. 82]
F.N. Tou, M.D. Williams, R.E. Fikes, A. Henderson, T.W. Malone, *RABBIT: An Intelligent Database Assistant*, Proceedings of AAAI-82, Second National Conference on Artificial Intelligence (Pittsburgh, PA), August 1982, pp. 314-318.

[Turner 88]
A.A. Turner (ed.), *Mental Models and User-Centered Design, Workshop Report (Breckenridge, CO)*, Institute of Cognitive Science, University of Colorado, Boulder, CO, Technical Report, No. 88-9, 1988.

[Walker, Kintsch 85]
W.H. Walker, W. Kintsch, *Automatic and Strategic Aspects of Knowledge Retrieval*, Cognitive Science, Vol. 9, 1985, pp. 261-283.

[Williams 84]
M.D. Williams, *What Makes RABBIT Run?*, International Journal of Man-Machine Studies, Vol. 21, 1984, pp. 333-352.

[Williams, Hollan 81]
M.D. Williams, J.D. Hollan, *The Process of Retrieval from Very Long Term Memory*, Cognitive Science, Vol. 5, 1981, pp. 87-119.